

**NAME**

llvm-lipo - LLVM tool for manipulating universal binaries

**SYNOPSIS**

**llvm-lipo** [*filenames...*] [*options*]

**DESCRIPTION**

**llvm-lipo** can create universal binaries from Mach-O files, extract regular object files from universal binaries, and display architecture information about both universal and regular files.

**COMMANDS**

**llvm-lipo** supports the following mutually exclusive commands:

**-help, -h**

Display usage information and exit.

**-version**

Display the version of this program.

**-verify\_arch <architecture 1> [<architecture 2> ...]**

Take a single input file and verify the specified architectures are present in the file. If so then exit with a status of 0 else exit with a status of 1.

**-archs**

Take a single input file and display the architectures present in the file. Each architecture is separated by a single whitespace. Unknown architectures are displayed as unknown(CPUtype,CPUsubtype).

**-info**

Take at least one input file and display the descriptions of each file. The descriptions include the filename and architecture types separated by whitespace. Universal binaries are grouped together first, followed by thin files. Architectures in the fat file: <filename> are: <architectures> Non-fat file: <filename> is architecture: <architecture>

**-thin**

Take a single universal binary input file and the thin flag followed by an architecture type. Require the output flag to be specified, and output a thin binary of the specified architecture.

**-create**

Take at least one input file and require the output flag to be specified. Output a universal binary

combining the input files.

**-replace**

Take a single universal binary input file and require the output flag to be specified. The replace flag is followed by an architecture type, and a thin input file. Output a universal binary with the specified architecture slice in the universal binary input replaced with the contents of the thin input file.

**-sealign**

Additional flag that can be specified with create and replace. The sealign flag is followed by an architecture type, and an alignment. The alignment is a hexadecimal number that is a power of 2. Output a file in which the slice with the specified architecture has the specified alignment.

**BUGS**

To report bugs, please visit <<https://github.com/llvm/llvm-project/issues/>>.

**AUTHOR**

Maintained by the LLVM Team (<https://llvm.org/>).

**COPYRIGHT**

2003-2023, LLVM Project